What is claimed is:

1. An external programmer for an implantable medical device, comprising:

a processor;

input and display means;

telemetry circuitry for communicating with an implanted device;

a nonvolatile memory for containing executable startup code;

a volatile system memory for containing executable operating system and application code;

a nonvolatile image storage medium;

a target state image saving routine for creating an image of a target state and storing the target state image as a target state data structure and a target state memory image in the image storage medium;

an operating system return routine for returning control of the system to the operating system software after execution of the target state image saving routine; and,

a restore routine for transitioning the system from a non-operational state to the target state by restoring the contents of volatile memory and processor registers to the target state in accordance with the target state image contained in the image storage medium and executing the operating system return routine.

.

2. The programmer of claim 1 wherein the target state image comprises a compressed copy of selected contents of nonvolatile memory in the target state.

3. The programmer of claim 1 wherein the image storage medium is flash ROM memory.

4. The programmer of claim 1 further comprising:

code in the target state image saving routine for storing the target state memory image as those pages in linear address space that are mapped into page frames of physical address space by a paging mechanism in the target state, and further wherein the target state data structure contains information derived from page tables defining the mapping of the page frames in the target state and values of processor registers in the target state; and,

code in the restore routine for loading processor registers with the values contained in the target state data structure and for storing the memory image in volatile memory in a manner that restores the page frames of the target state in accordance with the information derived from page tables contained in the target state data structure.

5. The programmer of claim 1 wherein the target state image contained in the image storage medium includes contents of video memory.

6. The programmer of claim 1 wherein further comprising startup code in the nonvolatile memory that initializes hardware registers with preset values prior to execution of the restore routine.

7. The programmer of claim 1 further comprising a target state process for moving the system to the target state before execution of the target state image saving routine.

8. The programmer of claim 4 wherein the operating system return routine is contained in the page frames of the target state.

9. The programmer of claim 1 wherein the operating system return routine is within an interrupt servicing routine of the operating system software.

10. The programmer of claim 1 wherein the image storage medium is a processor-executable storage medium and the restore routine is contained in the image storage medium.

11. The programmer of claim 4 wherein the retrieval and storing of the memory image by the restore routine is performed using physical addresses with paging switched off, and further wherein the restore routine switches on paging before transfer to the operating system return routine.

12. The programmer of claim 4 wherein the restore routine relocates itself during execution to memory locations in volatile memory that unused in the target state.

13. The programmer of claim 4 wherein the target state image saving routine scans the presently used paging directory and paging tables and generates a copy of the page directory filled with linear addresses of paging tables to enable translation of linear addresses to physical addresses.

14. The programmer of claim 13 wherein the target state image saving routine further generates a list of physical addresses of page frames stored in volatile memory locations from the contents of page tables.

15. The programmer of claim 13 wherein the target state image saving routine further generates a list of linear addresses of pages that are mapped to page frames in memory in the target state to enable translation of physical addresses to linear addresses by indexing into the list with a physical address.

16. The programmer of claim 15 wherein the target state image saving routine uses the list of linear addresses of mapped pages and the list of physical addresses of page frames to store the memory image in the image storage medium.

17. The programmer of claim 4 wherein the target state data structure includes a list of physical addresses that corresponds to volatile memory locations in which page frames are stored in the target state, the list being employed by the restore routine in storing the memory image retrieved from the storage device into volatile memory at selected locations.

18. The programmer of claim 4 wherein the restore routine retrieves the target state data structure from a known location in the image storage medium.

19. The programmer of claim 4 wherein the target state data structure contains the physical address corresponding to the memory location of the target state data structure in the target state, enabling the restore routine to write to the target state data structure.

20. The programmer of claim 4 wherein the target state data structure contains the physical address of the page directory in the target state, and further comprising code in the restore routine for: (a) patching the page directory so that physical addresses of restore routine instructions are mapped to the same physical addresses, (b) saving information for unpatching the page directory in the target state data structure, and (c) enabling paging before jumping to the operating system return routine.